



# CP2110 Errata

---

This document contains information on the CP2110 errata. The latest available revision of this device is revision F01-GM / F02-GM1.

For errata on older revisions, refer to the errata history section for the device. The revision information is typically specified in or near the trace code on the device. Refer to the package marking information in the data sheet for more information.

Errata effective date: October, 2020.

## 1. Active Errata Summary

These tables list all known errata for the CP2110 and all unresolved errata in revision F01-GM / F02-GM1 of the CP2110.

**Table 1.1. Errata History Overview**

Designator	Title/Problem	Exists on Revision:	
		Data Sheet revision 1.2 and earlier	F01/F02
CP2110_E101	ROM Programming Voltage	X	—
CP2110_E102	Improper Behavior of RS485 Ouput Pin at High Baud Rates	—	X

**Table 1.2. Active Errata Status Summary**

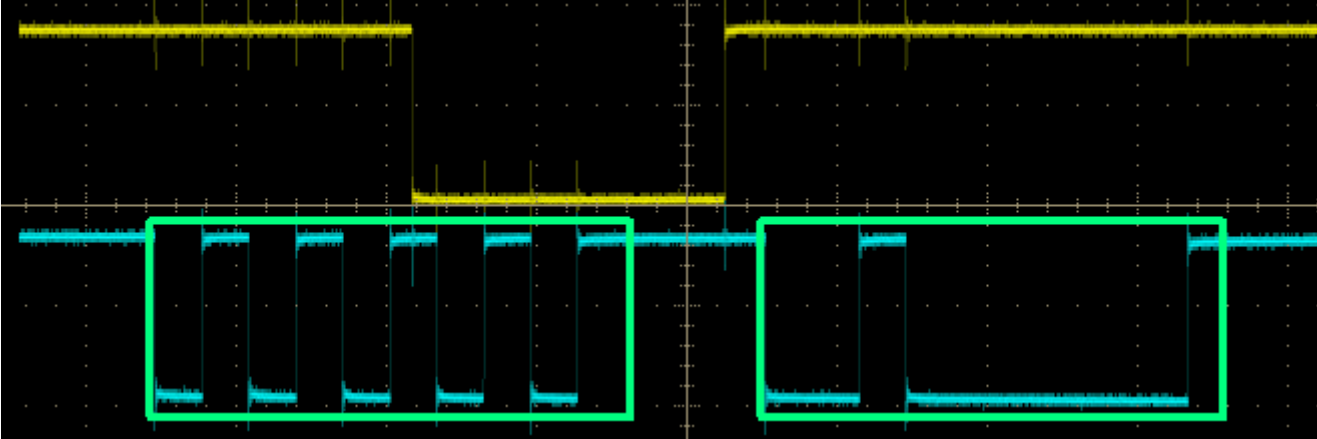
Errata #	Designator	Title/Problem	Workaround Exists	Affected Revision	Resolution
1	CP2110_E101	<a href="#">ROM Programming Voltage</a>	Yes	Data Sheet revision 1.2 and earlier	Data Sheet revision 1.3 or later
2	CP2110_E102	<a href="#">Improper Behavior of RS485 Ouput Pin at High Baud Rates</a>	Yes	F01/F02	—

## 2. Detailed Errata Descriptions

### 2.1 CP2110\_E101 – ROM Programming Voltage

<b>Description of Errata</b>
The data sheet incorrectly indicates that VDD must remain at 3.3 V or higher to successfully write to the configuration ROM. Instead, the voltage on the VIO pin must remain at 3.3 V or higher when writing to the configuration ROM.
<b>Affected Conditions / Impacts</b>
For systems that connect VDD and VIO together, there is no impact. For systems that have a separate voltage source for VIO and are configuring the ROM in-system, VIO must remain at 3.3 V while programming is in progress.
<b>Workaround</b>
For systems that connect VDD and VIO together, keep both power supplies above 3.3 V when programming. For systems that have a separate voltage source for VIO and are configuring the ROM in-system, VIO must remain at 3.3 V while programming is in progress.
<b>Resolution</b>
This issue will be resolved in data sheet revision 1.3 or later.

2.2 CP2110\_E102 – Improper Behavior of RS485 Ouput Pin at High Baud Rates

<b>Description of Errata</b>
<p>Under the following conditions, the 'GPIO.3_RS485' pin can be improperly de-asserted while serial data is being transmitted on the TX pin:</p> <ul style="list-style-type: none"><li>• The 'GPIO.3_RS485' pin is used to control an RS-485 transceiver (see Figure 9 of the CP2110 datasheet).</li><li>• The data block being transmitted is larger than 63 bytes (the maximum number of bytes in one USB packet).</li><li>• The baud rate is greater than 700 kbps.</li></ul> <p>In the following oscilloscope image, the top signal is the active-high RS485 pin and the bottom signal is the TX pin. The green boxes indicate the first (0x55) and second (0x02) data bytes of the second (or subsequent) group, including the start and stop bits. The image shows that the RS485 signal is improperly de-asserted while transmitting the first byte of the second (or subsequent) group of 63 bytes that correspond to one USB frame.</p> 
<b>Affected Conditions / Impacts</b>
De-assertion of the RS485 pin during active transmission results in corruption of the data being transmitted.
<b>Workaround</b>

Either of these two workarounds can be used to avoid the improper de-assertion of RS485:

- If transmit data block is larger than 63 bytes, do not use baud rates above 700 kbps.
- If baud rate > 700 kbps, do not send more than 60 bytes at a time.

The following application code can be used to break a large data block into multiple calls to the interface library HidUart\_Write() function.

```
HID_UART_STATUS WriteInBlocks(U8* buffer, U32 numToWrite, U32 blockSize, U32* numWritten)
{
    // Input parameters:
    //   buffer      : The buffer containing data to transmit
    //   numToWrite   : The total number of bytes to write (i.e. size of buffer)
    //   blockSize    : The number of bytes to write at a time (must be 60 or less)
    // Return parameters:
    //   numWritten   : Set to the total number of bytes written.

    HID_UART_STATUS status = HID_UART_SUCCESS;
    U32 totalBytesWritten = 0;
    U32 bytesWritten = 0;

    while (totalBytesWritten < numToWrite)
    {
        U32 bytesToWriteThisTime = numToWrite - totalBytesWritten;
        if (bytesToWriteThisTime > blockSize)
        {
            bytesToWriteThisTime = blockSize;
        }

        // Send the UART data to the device to transmit
        U8 * pBuf = buffer + totalBytesWritten;
        status = HidUart_Write(handle, pBuf, bytesToWriteThisTime, &bytesWritten);

        // Update number of bytes written
        totalBytesWritten += bytesWritten;

        if (status != HID_UART_SUCCESS)
        {
            // Error occurred; break out of loop
            break;
        }
    }

    *numWritten = totalBytesWritten;
    return status;
}
```

### Resolution

There is currently no resolution for this issue.

### 3. Errata History

This section contains the errata history for CP2110 devices.

For errata on the latest revision, refer to the beginning of this document. The device data sheet explains how to identify chip revision, either from package marking or electronically.

#### 3.1 Errata Summary

This table lists all errata for the CP2110.

**Table 3.1. Errata History Status Summary**

Errata #	Designator	Title/Problem	Workaround Exists	Affected Revision	Resolution
There are no errata in the errata history for this device.					

## 4. Revision History

### Revision 0.2

October, 2020

- Renamed D1 to [CP2110\\_E101](#).
- Added [CP2110\\_E102](#).

### Revision 0.1

August, 2012

- Initial release.

# Simplicity Studio

One-click access to MCU and wireless tools, documentation, software, source code libraries & more. Available for Windows, Mac and Linux!



**IoT Portfolio**

[www.silabs.com/IoT](http://www.silabs.com/IoT)



**SW/HW**

[www.silabs.com/simplicity](http://www.silabs.com/simplicity)



**Quality**

[www.silabs.com/quality](http://www.silabs.com/quality)



**Support & Community**

[www.silabs.com/community](http://www.silabs.com/community)

## Disclaimer

Silicon Labs intends to provide customers with the latest, accurate, and in-depth documentation of all peripherals and modules available for system and software implementers using or intending to use the Silicon Labs products. Characterization data, available modules and peripherals, memory sizes and memory addresses refer to each specific device, and "Typical" parameters provided can and do vary in different applications. Application examples described herein are for illustrative purposes only. Silicon Labs reserves the right to make changes without further notice to the product information, specifications, and descriptions herein, and does not give warranties as to the accuracy or completeness of the included information. Without prior notification, Silicon Labs may update product firmware during the manufacturing process for security or reliability reasons. Such changes will not alter the specifications or the performance of the product. Silicon Labs shall have no liability for the consequences of use of the information supplied in this document. This document does not imply or expressly grant any license to design or fabricate any integrated circuits. The products are not designed or authorized to be used within any FDA Class III devices, applications for which FDA premarket approval is required, or Life Support Systems without the specific written consent of Silicon Labs. A "Life Support System" is any product or system intended to support or sustain life and/or health, which, if it fails, can be reasonably expected to result in significant personal injury or death. Silicon Labs products are not designed or authorized for military applications. Silicon Labs products shall under no circumstances be used in weapons of mass destruction including (but not limited to) nuclear, biological or chemical weapons, or missiles capable of delivering such weapons. Silicon Labs disclaims all express and implied warranties and shall not be responsible or liable for any injuries or damages related to use of a Silicon Labs product in such unauthorized applications.

## Trademark Information

Silicon Laboratories Inc.®, Silicon Laboratories®, Silicon Labs®, SiLabs® and the Silicon Labs logo®, Bluegiga®, Bluegiga Logo®, ClockBuilder®, CMEMS®, DSPLL®, EFM®, EFM32®, EFR, Ember®, Energy Micro, Energy Micro logo and combinations thereof, "the world's most energy friendly microcontrollers", Ember®, EZLink®, EZRadio®, EZRadioPRO®, Gecko®, Gecko OS, Gecko OS Studio, ISOModem®, Precision32®, ProSLIC®, Simplicity Studio®, SiPHY®, Telegesis, the Telegesis Logo®, USBXpress®, Zentri, the Zentri logo and Zentri DMS, Z-Wave®, and others are trademarks or registered trademarks of Silicon Labs. ARM, CORTEX, Cortex-M3 and THUMB are trademarks or registered trademarks of ARM Holdings. Keil is a registered trademark of ARM Limited. Wi-Fi is a registered trademark of the Wi-Fi Alliance. All other products or brand names mentioned herein are trademarks of their respective holders.



Silicon Laboratories Inc.  
400 West Cesar Chavez  
Austin, TX 78701  
USA

<http://www.silabs.com>